Assignment 5

Yifei Gu

260906596

Part a)

To build the noise model, I did windowing the data with **Hann Window** to **smooth the data first**. Because it has a long flat period in the middle. I also tried **Welch's method** to smooth the data. It **does a better job than Hann Window**, but Welch's method **reduces the resolution** of the data and I had **difficulties to interpolate** the smoothed data to the equal number of points with the original data. Thus, I chose **Hann Window**.

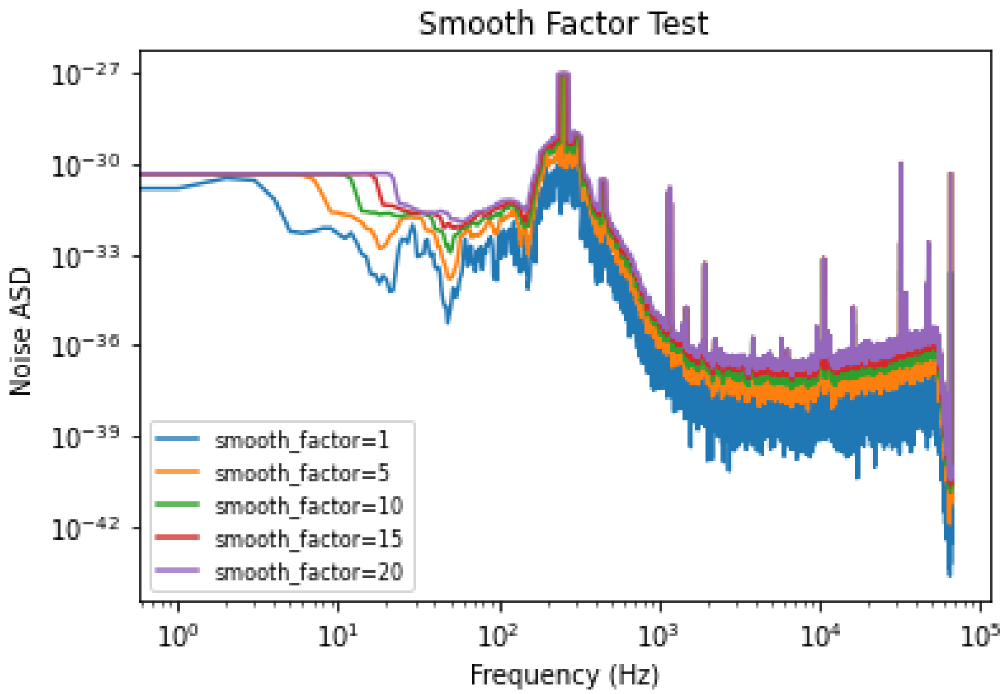
```
def noise_model(signal,smooth_factor):
    #I also tried welch's method but it couldn't work
    #split the each signal into 8 segments
    #segment_length=len(signal)/8
    # Using Welch's Method to get a smoothed psd
    # The window chose here is hann
    #freqs, psd = sig.welch(signal,window="hann",nperseg=segment_length)

    # Using hann windowing to smooth the signal
    window = sig.get_window('hann',len(signal))
    signal_win = signal*window
    sft = np.fft.rfft(signal_win)
    N = np.abs(sft)**2
    N_smooth=smooth(N,smooth_factor)
    N = np.maximum(N,N_smooth)
    return N
```

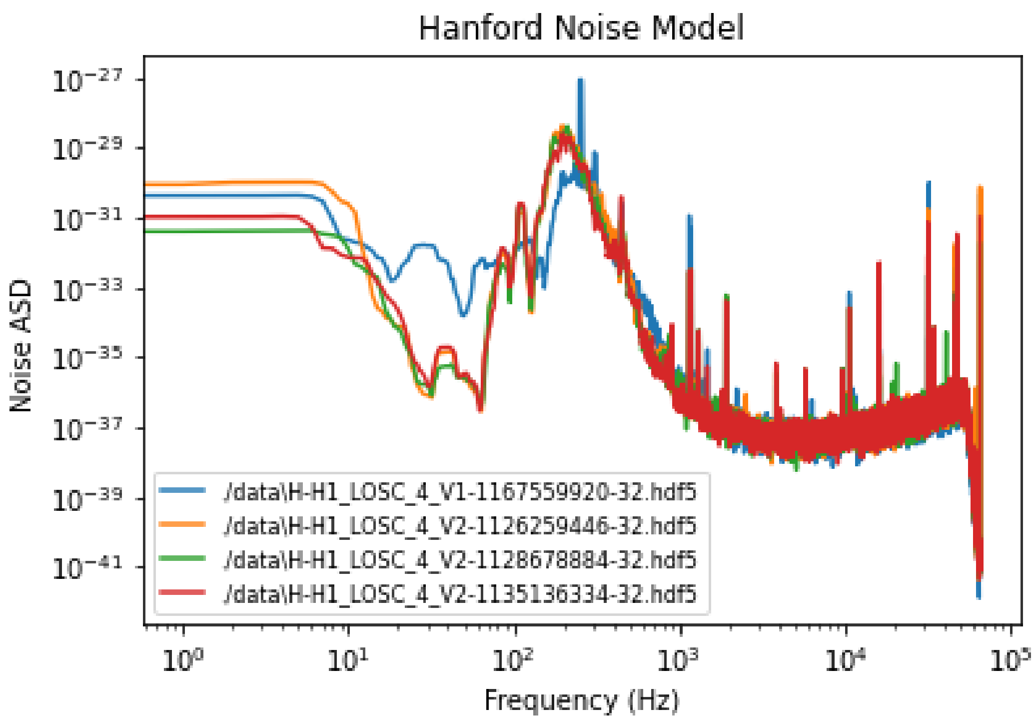
The noise in frequency domain was **taken as the square** of the **strain** in frequency. I also **smoothed the noise model** by convolution with a box car function:

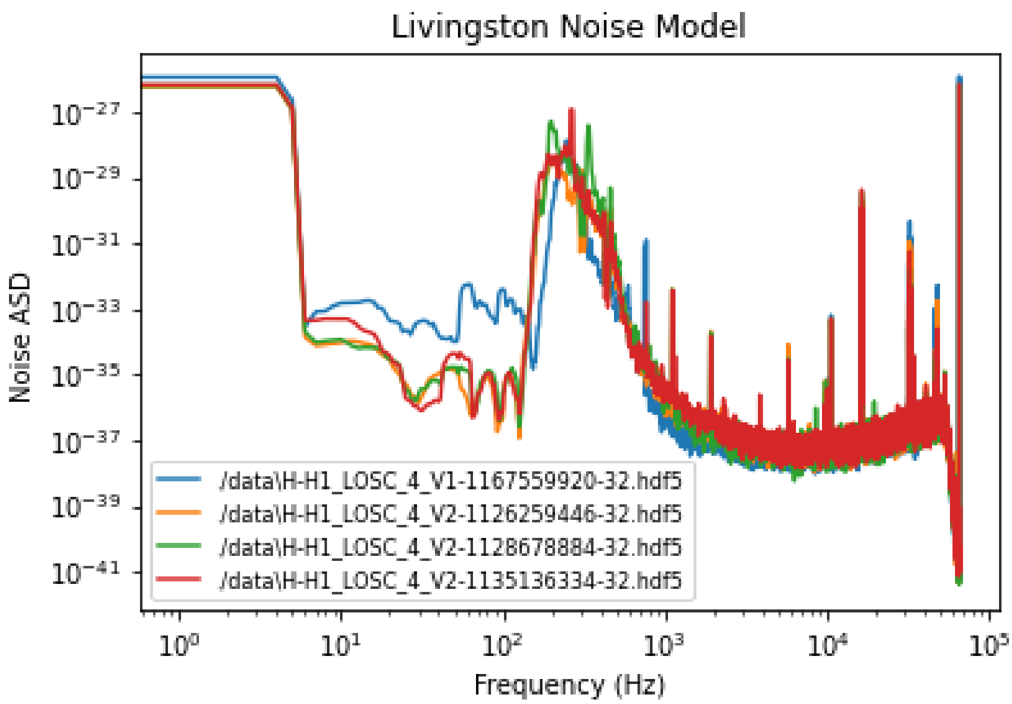
```
def smooth(a,n):
    vec = np.zeros(len(a))
    vec[:n]=1
    vec[-n:]=1
    fft1 = np.fft.rfft(a)
    fft2 = np.fft.rfft(vec)
    return np.fft.irfft((fft1*fft2),len(a))
```

I did a test with box cars in different length. Here is how the noise model with different smooth factors performs:



Based on this test, **a box car with width (I call it smooth factor in the code) of 5 was chosen** as it **removed sharp peaks but avoid flattening features.** Noise models for Hanford and Livingston are shown below:





Part b)

Since I windowed the data, in the matched filter I windowed the templates as well.

```
def match_filter(strain, template, dt,smooth_factor):
    #window both the signal and strain with hann window
    window = sig.get_window('hann',len(strain))
    sft = np.fft.rfft(strain*window)
    tft = np.fft.rfft(th*window)
    freq = np.fft.fftfreq(len(window),dt)
    #noise
    noise = noise_model(strain,smooth_factor)
    # Do the matched filter
    mf_ft = np.conj(tft)*(sft/noise)
    mf = np.fft.irfft(mf_ft)
    # Get freq spacing
    df = freq[1]-freq[0]
    int = intg.cumtrapz(np.abs(mf_ft), dx=df, initial=0)
    mid_idx = np.argmin(np.abs(int - max(int)/2))
    return mf, mf_ft, mid_idx, noise
```

After loading all the data, I sorted them as which detector each collected from. Searching through all data with corresponding templates, I found a few events. They are attached in the end of the document with all the other questions.

Part c)

In this question, I defined the noise as the standard deviation of the matched filter. The SNR of scatter was calculated as the ratio between the maximum of the matched filter to the noise.

```
def SNR_scatter(mf):
    SNR = np.max(np.abs(mf))/np.std(mf)
    return SNR
```

Part d)

The analytic SNR, I calculated the noise in the matched filter first in frequency domain then transform back to time domain and take square root of it:

$$\sigma = \sqrt{\frac{template^{*}(f)template(f)}{noise\ model}}$$

The SNR was estimated as the maximum of the ratio between matched filter signal and the noise:

$$SNR = \max\left(\frac{Matched\ filter\ signal}{\sigma}\right)$$

Where 'Matched filter signal' is the raw output of the matched filter in the time domain. Here is how I built the function for calculating the analytic SNR:

```python
def SNR_analytic(template,noise,mf):

    #  fft of template
    t_ft = np.fft.rfft(template)
    rhs = mf
    #  calcualte sigma
    lhs_ft= np.conj(t_ft)*(t_ft/noise)
    lhs= np.fft.irfft(lhs_ft)
    N = np.sqrt(np.abs(lhs))
    SNR = np.max(np.abs(rhs/N))
    return SNR
```

There are differences between the SNR from scatter and the analytic SNR. This could arise from the windowing of the data, which decreases the value of the sigma of the noise std and increases the SNR value.

Part e)

The output of the matched filter in the frequency domain is the weighting of correspond frequency. By taking the cumulative integral of the matched filter, and finding the frequency at which the cumulative integral is half the full integral, we can get the mid frequency.

$$\int_0^{f'} F(f)df = \frac{1}{2}\int_0^{\infty} F(f)df$$

Here is the section that finds the mid frequency in my matched filter:

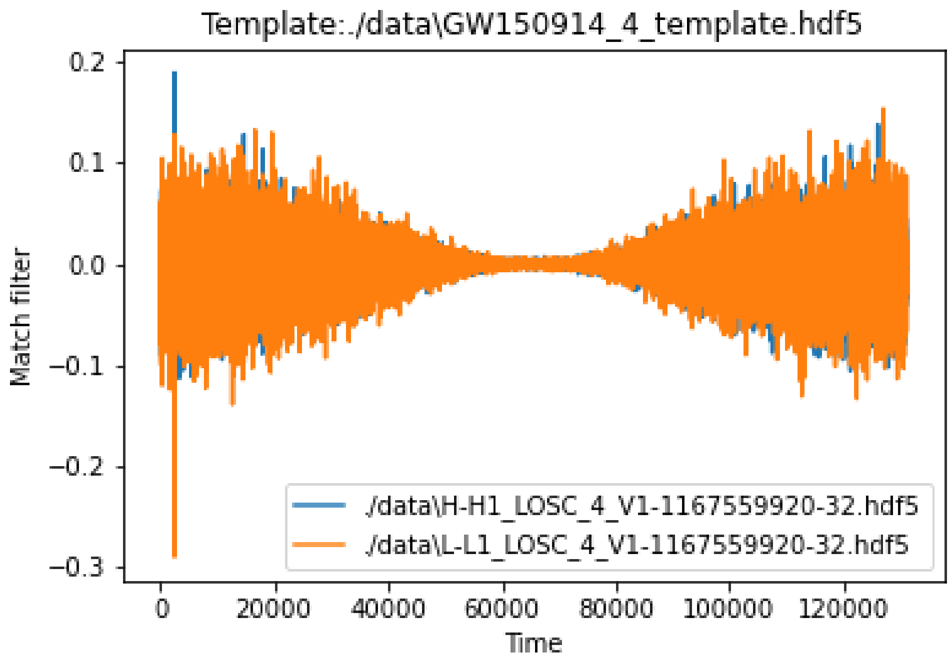
```python
df = freq[1]-freq[0]
int = intg.cumtrapz(np.abs(mf_ft), dx=df, initial=0)
mid_idx = np.argmin(np.abs(int - max(int)/2))
return mf, mf_ft, mid_idx, noise
```

Part f)

I take the error on the timing by taking the FWHM of the peaks in the matched filter:

$$\sigma = \frac{FWHM}{2.355}$$

Results and plots:



The SNR of Hanford from scatter:
9.19110383637002

The SNR of Livingston from scatter:
11.408889370342004

The SNR of combined Hanford and Livingston from scatter:
14.650568125353995

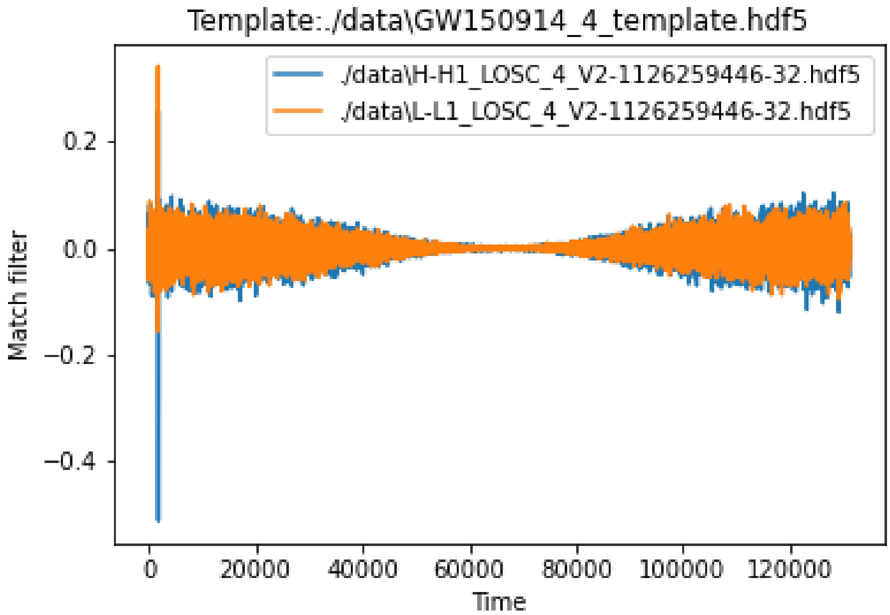
The analytic SNR of Hanford:
4.379836751466547

The analytic SNR of Livingston with noise model:
14.32572612118479

The analytic SNR of combined Hanford and Livingston with noise model:
14.980300359762259

The Mid frequency of Hanford:
4011

The Mid frequency of Liveingston:
3658



The SNR of Hanford from scatter:
25.342041768947347

The SNR of Livingston from scatter:
19.358192980170866

The SNR of combined Hanford and Livingston from scatter:
31.889790160435496

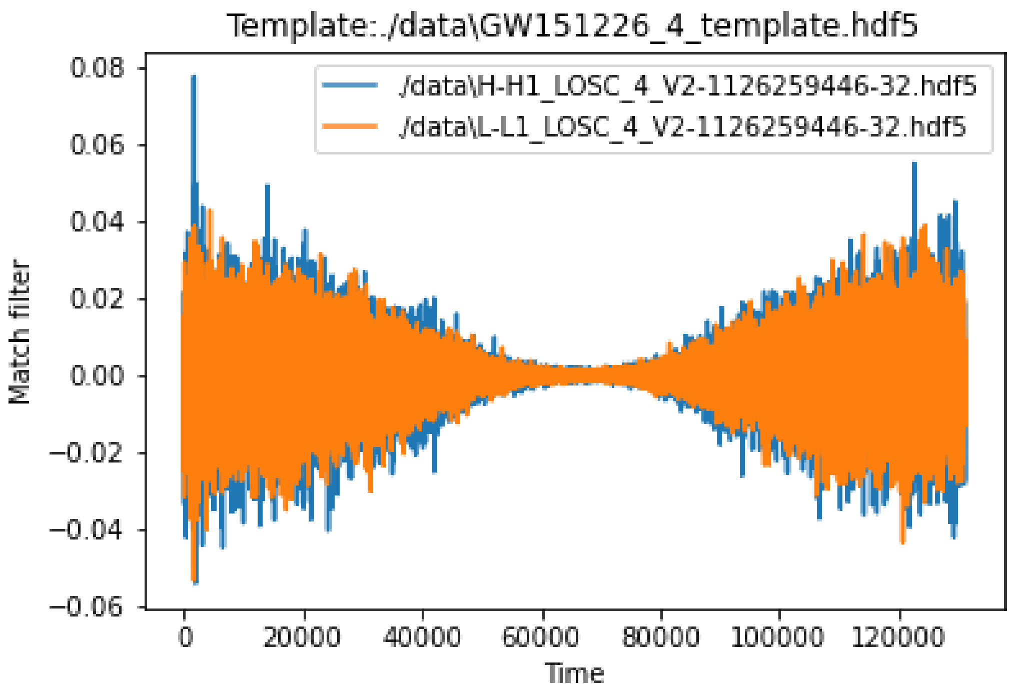
The analytic SNR of Hanford:
4.200687668365736

The analytic SNR of Livingston with noise model:
6.244003655335075

The analytic SNR of combined Hanford and Livingston with noise model:
7.5255138386025004

The Mid frequency of Hanford:
3970

The Mid frequency of Liveingston:
4267



The SNR of Hanford from scatter:
9.391907887600901

The SNR of Livingston from scatter:
7.430506743792907

The SNR of combined Hanford and Livingston from scatter:
11.975824156972742

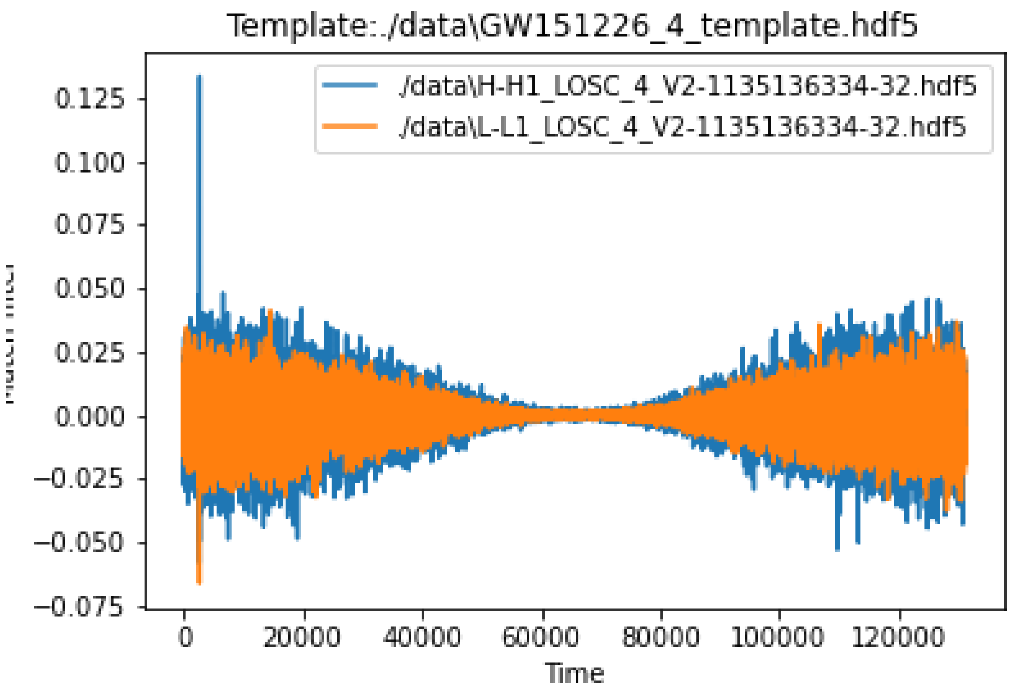
The analytic SNR of Hanford:
9.89912109425204

The analytic SNR of Livingston with noise model:
6.91223556828486

The analytic SNR of combined Hanford and Livingston with noise model:
12.073590973282474

The Mid frequency of Hanford:
4538

The Mid frequency of Liveingston:
5156



The SNR of Hanford from scatter:
14.311086740055122

The SNR of Livingston from scatter:
10.284518516138242

The SNR of combined Hanford and Livingston from scatter:
17.623238203865142

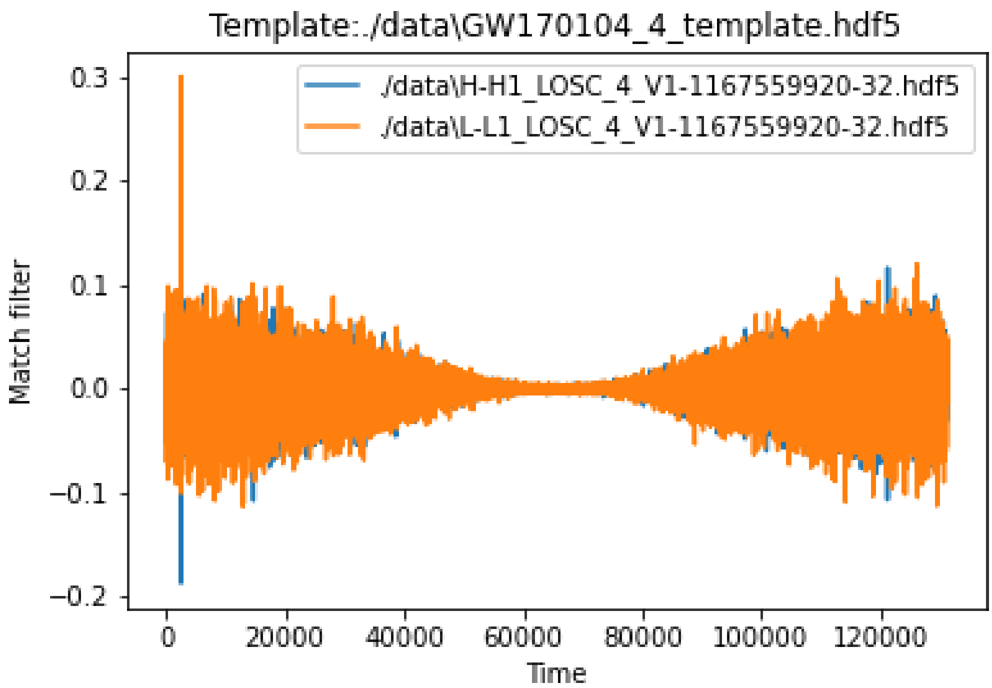
The analytic SNR of Hanford:
7.988568134349277

The analytic SNR of Livingston with noise model:
6.203323662691908

The analytic SNR of combined Hanford and Livingston with noise model:
10.114269390383768

The Mid frequency of Hanford:
4213

The Mid frequency of Liveingston:
5503



The SNR of Hanford from scatter:
11.020541940114109

The SNR of Livingston from scatter:
14.206050222326628

The SNR of combined Hanford and Livingston from scatter:
17.979549704402512

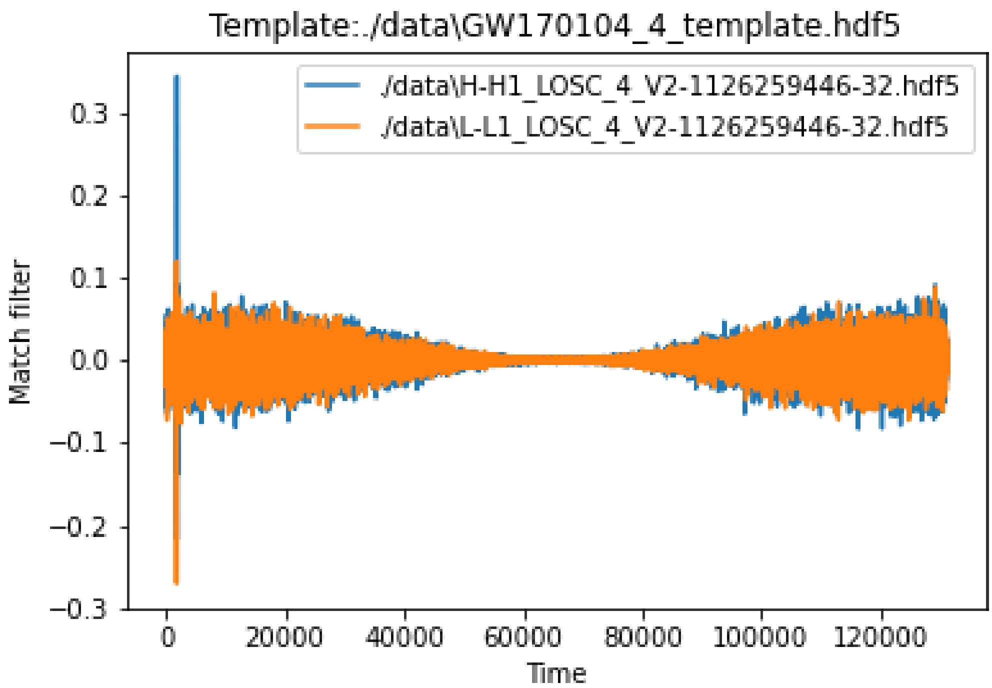
The analytic SNR of Hanford:
6.567411303849463

The analytic SNR of Livingston with noise model:
4.791138843201365

The analytic SNR of combined Hanford and Livingston with noise model:
8.129323628000217

The Mid frequency of Hanford:
3976

The Mid frequency of Liveingston:
3625



The SNR of Hanford from scatter:
20.782707334251796

The SNR of Livingston from scatter:
18.775223257688705

The SNR of combined Hanford and Livingston from scatter:
28.007676314132492

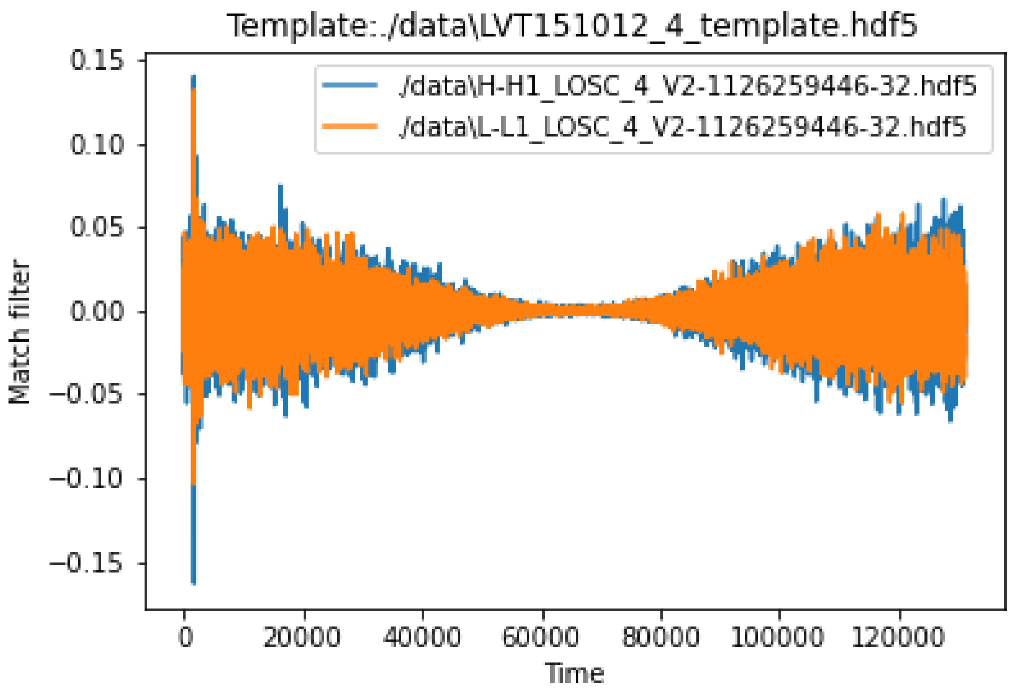
The analytic SNR of Hanford:
14.082387315376844

The analytic SNR of Livingston with noise model:
3.918541145429638

The analytic SNR of combined Hanford and Livingston with noise model:
14.61740733538994

The Mid frequency of Hanford:
3933

The Mid frequency of Liveingston:
4309



The SNR of Hanford from scatter:
12.683917707505566

The SNR of Livingston from scatter:
11.88748273025778

The SNR of combined Hanford and Livingston from scatter:
17.38372842841691

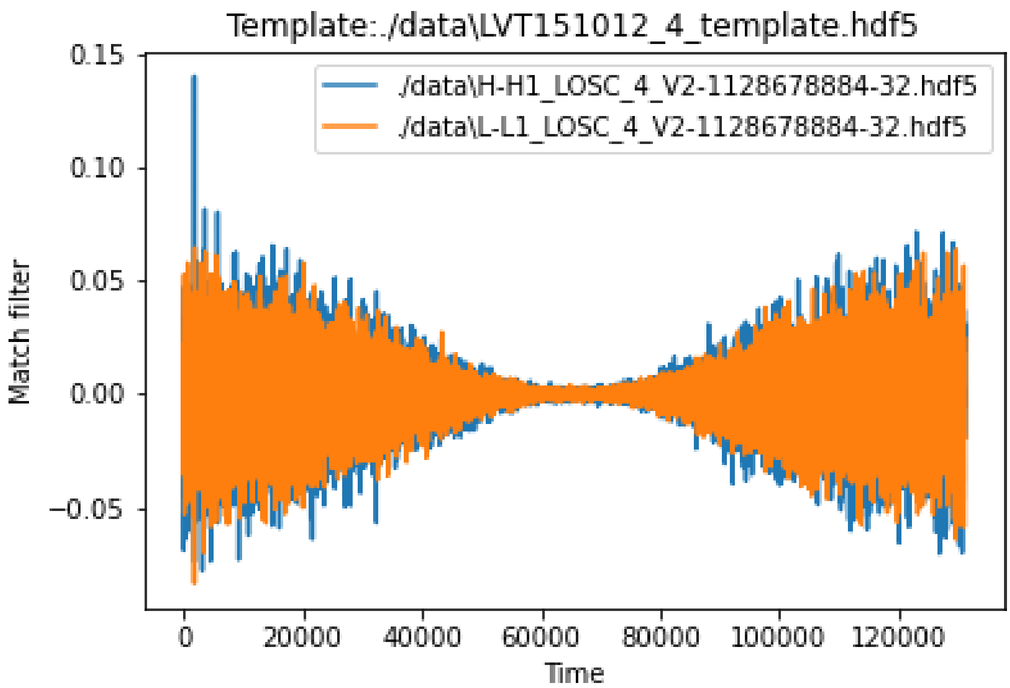
The analytic SNR of Hanford:
7.990765025144486

The analytic SNR of Livingston with noise model:
7.8148390552146125

The analytic SNR of combined Hanford and Livingston with noise model:
11.176942119648826

The Mid frequency of Hanford:
3775

The Mid frequency of Liveingston:
4211



The SNR of Hanford from scatter:
10.239040699192035

The SNR of Livingston from scatter:
6.907763995756118

The SNR of combined Hanford and Livingston from scatter:
12.35132210983

The analytic SNR of Hanford:
10.398137690848174

The analytic SNR of Livingston with noise model:
6.822272318893268

The analytic SNR of combined Hanford and Livingston with noise model:
12.43642501006518

The Mid frequency of Hanford:
3646

The Mid frequency of Liveingston:
4159